








Reto 6: Distribución de vacunas - Corporación Umbrella

 Propósito	Vamos a solucionar un problema de lectura de archivos y visualización de la información contenida con el fin de comparar reportes de distribución
 Dificultad	
 Fecha	@May 23, 2022
 Fuente	MisionTic2022 Ciclo 1
 Numero	25
 Adjuntos	Reto_Semana_6_23may2022.pdf

Autor: **David Santiago Urbano Rivadeneira**

Proceso Ideal

1. **Identificar** el problema

- ¿Cual es el problema? → Desarrollar una interfaz que permita mostrar los indicadores de desempeño de cada vehiculo, para cada cliente
- Stakeholders
 - Conductores de Transportes del Norte. (clientes)
 - Departamento de logistica
- Objetivos: se detallan en los requerimientos funcionales abajo

Restricciones 

Restricciones

1. Los datos serán suministrados en dos archivos .csv

2. Definir el problema

- ¿Que conozco? se ingresarán 2 planillas, Las planillas:
 - *ValoresAsignados.csv* y *ValoresRegistrados.csv* que se ingresará al programa consta de 3 variables (*Punto de distribución*, *Cajas entregadas* y *Tiempo de despacho(minutos)*).
- Se dividió el problema en los subproblemas siguientes
 - Leer desde el archivo “ValoresAsignados.csv” sus variables correspondientes
 - Leer desde el archivo “ValoresRegistrados.csv” sus variables correspondientes
 - Calcular los 5 indicadores (*Eficiencia en tiempos de despacho (%)*, *Tasa de entrega (cajas/min)*, *Nivel de cumplimiento de los despachos (%)*, *Entregas a tiempo (%)*) para posteriormente imprimirlos en pantalla dependiendo de el punto selecciona de distribución
 - Imprimir todos los puntos de distribución que registran ambas condiciones negativas: tanto sobre-entregas, como demora en su tiempo de despacho.
 - Registrar (escribir) en un archivo CSV (llamado *'registro_estadisticas.csv'*) para todos los puntos de distribución con sus 4 indicadores *Eficiencia en tiempos de despacho (%)*, *Tasa de entrega (cajas/min)*, *Nivel de cumplimiento de los despachos (%)*
 - Realizar 6 graficas *Carga_cajas(EjeY) vs PuntoDistribucion(EjeX)*, *Tiempo_minutos(EjeY) vs PuntoDistribucion(EjeX)*, Única gráfica que muestre *i y ii*

3. Estrategia

Ejemplos:

Entrada del programa:

```
5 #punto_distribucion=int(input())
ValoresAsignados.csv
ValoresRegistrados.csv
```

Salida esperada del programa:

```
Eficiencia en tiempos de despacho = 7.3 %
Tasa de entrega = 22.3 cajas/min
Nivel de cumplimiento de los despachos = 83.3 %
Entregas a tiempo = 50.0 %
```

4. Algoritmos

Requerimientos funcionales

El aplicativo debera desplegar los siguientes indicadores de desempeño para cualquier punto de distribución:

- Eficiencia en tiempos de despacho (%) = $100 * (\text{Tiempo total de despacho asignado} - \text{Tiempo total de despacho registrado}) / \text{Tiempo de total despacho asignado}$
- Tasa de entrega (cajas/min) = $\text{Cantidad total de cajas despachadas} / \text{Tiempo total de despacho}$
- Nivel de cumplimiento de los despachos (%) = $100 * (\text{Total de cajas despachadas} / \text{Total de cajas asignadas})$
- Entregas a tiempo (%) = $100 * (\text{No. de entregas a tiempo} / \text{No. Total de entregas realizadas})$.

Nota: este es un resultado que es constante, es decir, es independiente del punto de distribución porque toma en cuenta todos los puntos de distribución.

El aplicativo debera calcular la diferencia de tiempos de despacho, reflejando “lo asignado” menos “lo registrado”. Los valores negativos corresponden a demoras en el punto de distribución.

El aplicativo debera calcular la eficiencia en los tiempos de despacho, reflejada como un porcentaje que toma como referencia “lo asignado”.

Divide y venceras

- Main:
 1. Llamar las funciones
 2. Imprimir los valores solicitados
 3. para las graficas utilizar matplotlib
- Funcion 1 Control:

1. Almacenar los datos para cada uno de los archivos, recorriendo cada fila y guardandolos en una nueva matriz.
2. Definir las variables necesarias para el calculo de los indicadores, y sus respectivas formulas
3. Realizar el conteo de cajas entregadas a tiempo

Pseudocodigo para el conteo

Nombre: Cajas entregadas a tiempo

“Este algoritmo realiza el conteo de cajas entregadas a tiempo”

INICIO

Variables:

Lista de listas: `tabla_registrados`, `tabla_asignados`

Variable acumuladora: `num_entregas_atiempo`

Recibir: `tabla_registrados`, `tabla_asignados`

```
num_entregas_atiempo = 0
```

```
Para i ← 0 hasta longitud(lista_tuplas_asignado):
```

```
    Si tabla_asignados[i][2] < tabla_registrados[i][2] entoces:
```

```
        num_entregas_atiempo = num_entregas_atiempo + 1
```

FIN

4. Con los valores anteriores, definimos el iterador en este caso el punto_distribucion ingresado por el usuario. de la siguiente forma:

	0	1	2	3
	PuntoDistribucion	EficienciaTiemposDespacho_%	TasaEntrega_cajas/min	NivelCumplimientoDespachos_%
0				
1				
2				
3				
4				
5				
6				
7				
iterador				

Representa el recorrido de la matriz resultado, para la impresion de los indicadores.

5. Se realiza el registro de las estadísticas para todos los puntos de distribución.
6. Se muestran las gráficas solicitadas, y se guardan respectivamente

Código de la solución



Siguiendo las **convenciones estilísticas** de la PEP8 para escribir código en Python más legible

Código Main.py

```
import control as ctr

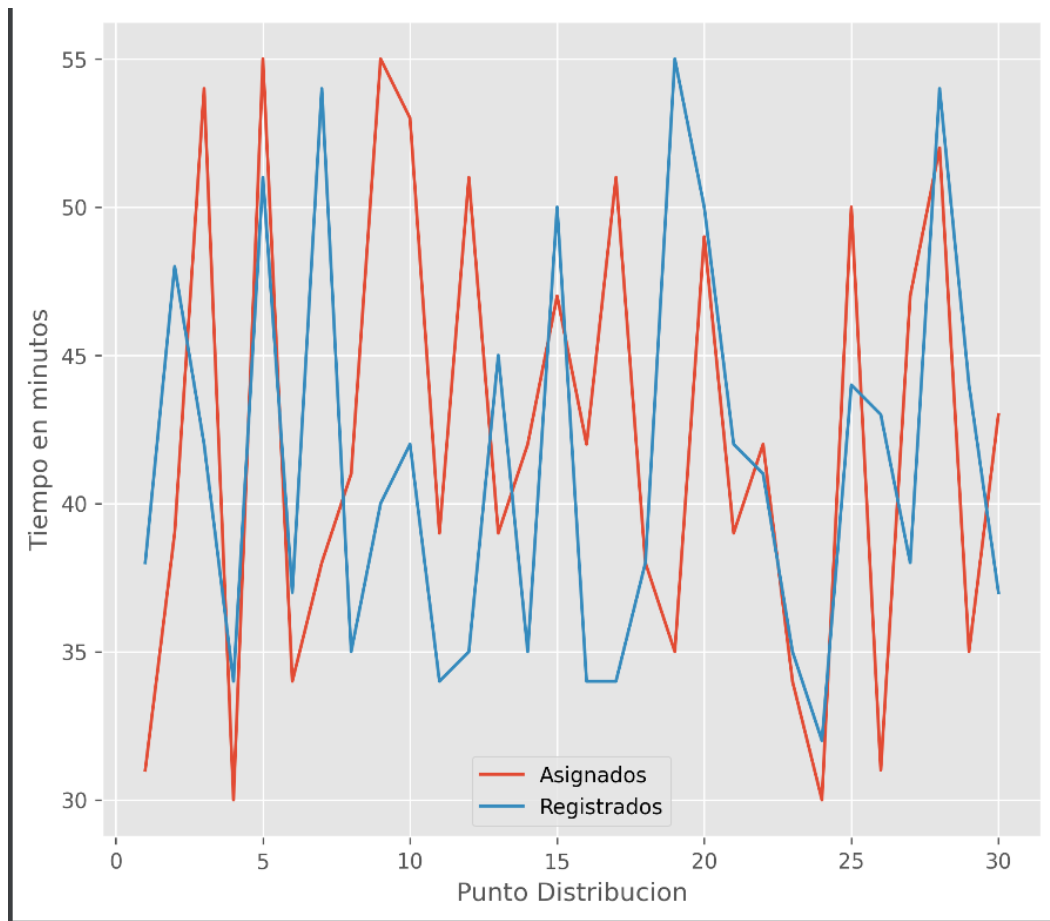
punto_distribucion = int(input("punto_distribucion="))

if __name__ == '__main__':
    ctr.leer_archivos(punto_distribucion)
```

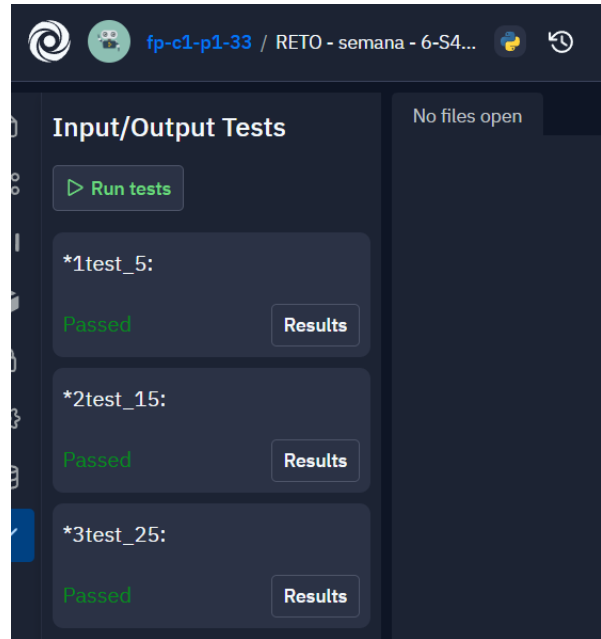
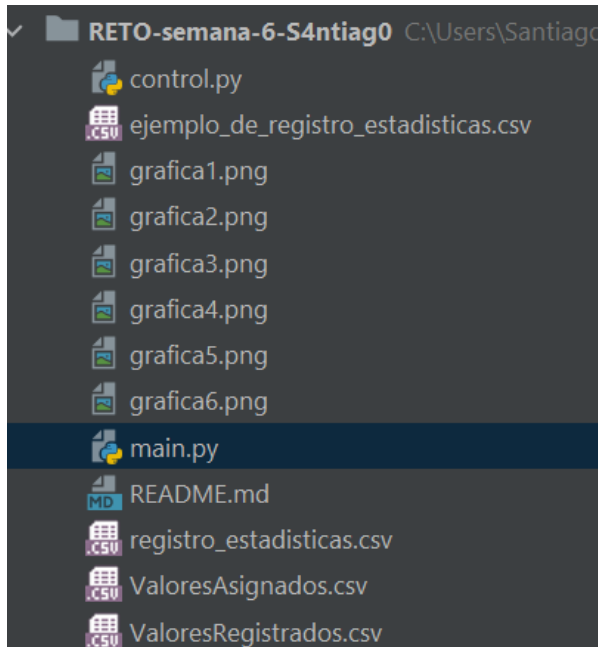
Código Main.py

[illegible]

Pruebas



ej Grafico#6: Comparativa entre tiempo de despacho Asignados y Registrados



[Link notion](#)